CS615 Endsem Exam (Spring 2019)

Max marks: 60 **Time: 180 mins**

- *Be brief, complete and stick to what has been asked.*
- *Unless asked for explicitly, you can cite results/proofs covered in class.*
- ***If you need to make any assumptions, state them clearly.***
- ***Read the question paper carefully before answering questions.***
- ***Do not copy solutions from others. Penalty for offenders: FR grade.***

1. *[5+5+5 marks]* Consider the predicates $D_1(p)$ and $D_2(p)$ defined below in separation logic.

$$\begin{aligned} D_1(p) \quad \equiv \quad & (p \mapsto [l : \mathsf{nil}, r : \mathsf{nil}]) \ \vee \\ & (\exists t\, (p \mapsto [l : \mathsf{nil}, r : t]) \star D_1(t)) \ \vee \\ & (\exists t\, (p \mapsto [l : t, r : \mathsf{nil}]) \star D_2(t)) \end{aligned}$$

$$\begin{aligned} D_2(p) \quad \equiv \quad & (p \mapsto [l : \mathsf{nil}, r : \mathsf{nil}]) \ \vee \\ & (\exists t_1, t_2\, (p \mapsto [l : t_1, r : t_2]) \star D_2(t_1) \star D_1(t_2)) \ \vee \\ & (\exists t\, (p \mapsto [l : t, r : \mathsf{nil}]) \star D_2(t)) \end{aligned}$$

   Give one model (using as few heap cells as possible) for each of the following separation logic formulas. I

   (a) $D_1(p) \wedge \neg D_2(p)$
   (b) $D_2(p) \wedge \neg D_1(p)$
   (c) $D_1(p) \mathrel{-\!\!\star} D_2(p)$

   In each case, you must provide an explanation of why your model satisfies the given formula. If you think a formula is unsatisfiable, you must provide reasons for this.

2. *[5 + 5 + 5 marks]* Let $\mathcal{A} = (A, \sqsubseteq, \sqcup, \sqcap, \top, \bot, \nabla)$ be an abstract domain, with $\alpha_{\mathcal{A}}$ and $\gamma_{\mathcal{A}}$ being the corresponding abstraction and concretization functions. Assume that $(A, \sqsubseteq, \sqcup, \sqcap, \top, \bot)$ is a complete lattice. We wish to design a new domain, called $\wp(\mathcal{A})$, that allows us to reason about sets of non-subsumed elements of A. Thus, if X is an element of $\wp(\mathcal{A})$, then $X \subseteq A$ and for every pair of distinct elements $x_1, x_2 \in X$, we have $x_1 \not\sqsubseteq x_2$. The abstraction and concretization functions for $\wp(\mathcal{A})$ are defined as follows: $\alpha_{\wp(\mathcal{A})}(S) = \{\alpha_{\mathcal{A}}(S)\}$ for all subset S of concrete states, and $\gamma_{\wp(\mathcal{A})}(X) = \cup_{x \in X}\ \gamma_{\mathcal{A}}(x)$.

   (a) A student defines the ordering relation $\sqsubseteq_{\wp(\mathcal{A})}$ for $\wp(\mathcal{A})$ as follows: For $X, Y \subseteq A$, $X \sqsubseteq_{\wp(\mathcal{A})} Y$ iff $X \subseteq Y$. Does this define a partial order on $\wp(\mathcal{A})$? Either give a proof or give a concrete counterexample.
   (b) The $\sqcap_{\wp(\mathcal{A})}$ operator can be defined simply as set intersection, i.e. for $X, Y \subseteq A$, $X \sqcap_{\wp(\mathcal{A})}\ Y = X \cap Y$. Show by means of an example that the $\sqcup_{\wp(\mathcal{A})}$ operator, however, cannot be defiend in general simply as set union.
   (c) Give as best a definition (even a procedural definition is ok) of $\nabla_{\wp(\mathcal{A})}$ as you can, using the operators of the abstract domain $\mathcal{A}$.

3. *[5+5+5+5 marks]* Suppose you are given that the following Hoare triples are valid:

   - $\{\varphi_1\}\ P\ \{\varphi_2\}$
   - $\{\neg\varphi_1\}\ P\ \{\neg\varphi_2\}$
   - $\{\varphi_3\}\ P\ \{\varphi_4\}$
   - $\{\neg\varphi_3\}\ P\ \{\neg\varphi_4\}$

   Explain with reasons if the following are true/valid. In each case, either give a proof or provide a concrete counterexample.

(a) φ_2 is the strongest post-condition of φ_1 with respect to the program P.

(b) φ_3 is the weakest pre-condition of φ_4 with respect to the program P.

(c) $\varphi_1 \leftrightarrow \varphi_3$

(d) $(\varphi_1 \rightarrow \varphi_3) \rightarrow (\varphi_2 \rightarrow \varphi_4)$

4. *[15 marks]* Consider the program given below:

```
int x, y, z;

while (x != 0) {
  x = x - 1;
  y = y + z;
  z = y + z;
}
```

We wish to analyze this program using the interval domain and find a loop invariant. You are told that the pre-condition is $(0 \leq x \leq 10) \wedge (10 \leq y \leq 20) \wedge (20 \leq z \leq 30)$.

Show how you arrive at the loop invariant in the interval domain by widening after two iterations of the loop. You must not widen in the first and second iterations – i.e. you must use the *lub* and not the widen operator the first two times you try to combine abstract states at the loop head to obtain an under-approximation of the loop invariant.